REMARKS

Claims 1-25 are currently pending in the subject application and are presently under consideration. Claim 10 has been amended as shown on p. 4 of the Reply. In addition, the specification has been amended as indicated on p. 2.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Objection to the Specification

The specification has been amended to indicate the use of trademarks JAVA and VISUAL BASIC.NET as indicated on p. 2. The use of these terms in both cases as examples of the respective generic terminology described in the specification is believed to satisfy the requirement that generic terminology accompany the use of trademarks in a specification as required by MPEP § 608.01(v). Accordingly, JAVA is used as an example of one of "[m]any programming languages . . . [that] catch or prevent many programming error(s) through compile-time checks and/or automatic memory management." VISUAL BASIC.NET is used as an example of "code created in language(s) that compile to the Common Language Runtime." Furthermore, it is believed that such use does not affect the validity of the respective trademarks. Accordingly, reconsideration and withdrawal of this objection is respectfully requested in view of the comments and amendments above.

II. Double Patenting

Claims 1, 15-17 and 19-25 is rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 12, 13, 15, 16, and 21 of co-pending U.S. Patent Application 10/681,789. However, applicants' Terminal Disclaimer is being filed concurrently herewith in the present application over co-pending U.S. Patent Application 10/681,789, which is believed to render this rejection moot. It is also noted that a corresponding Terminal Disclaimer was filed in co-pending U.S. Patent Application 10/681,789 on March 23, 2007 in view of the present application. Reconsideration and withdrawal of the rejection under the judicially created doctrine of

obviousness-type double patenting is respectfully requested in view of Applicants' current Terminal Disclaimer.

III. Rejection of Claims 1-25 Under 35 U.S.C. § 101

Claims 1-25 stand rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter. Claims 1, 15, 17, 20, and 22-25 are the independent claims. This rejection should be withdrawn for at least the following reasons. The Federal Circuit has clearly established in *Eolas Techs., Inc. v. Microsoft Corp.*, 399 F.3d 1325, 1338 (Fed. Cir. 2005) and *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352, 1358. (Fed.Cir. 1999) that inventions such as that claimed by applicants are statutory.

> This court must also decide whether software code made in the United States and exported abroad is a "component of a patented invention" under 271(f)... Section 271(f) refers to "components of a patented invention."... Title 35, section 101, explains that an invention includes "any new and useful process, machine, manufacture or composition of matter."... Without question, ***software code alone qualifies as an invention eligible for patenting under these categories***, at least as processes. *Eolas Techs., Inc. v. Microsoft Corp.*, 399 F.3d 1325, 1338 (Fed. Cir. 2005). (Emphasis added).

The Examiner incorrectly contends that the claimed subject matter is non-statutory because it could be construed as being software alone, not embodied in a computer readable medium. Applicants' representative disagrees with the Examiner's contention and submits that the Examiner is misconstruing the requirements necessary to fulfill the conditions for patentability under 35 U.S.C. § 101. The Federal Circuit in *Eolas Techs., Inc. v. Microsoft Corp.* clearly established that software code alone is statutory subject matter. Claims 1-25 claim executable code check systems or methods. Systems and methods by themselves are statutory subject matter. By the standards set forth in the above decision, a computer implemented system in the form of software,

hardware, or the combination of both clearly falls within the categories of statutory subject matter.

Furthermore, the subject claims produce a useful, concrete, and tangible result.

> Because the claimed process [method] applies the Boolean principle to produce a useful, concrete, tangible result … on its face the claimed process comfortably falls within the scope of §101. *AT&T Corp. v. Excel Communications, Inc.*, 172 F.3d 1352, 1358. (Fed.Cir. 1999); *See State Street Bank & Trust Co. v. Signature Fin. Group, Inc.*, 149 F.3d 1368, 1373, 47 USPQ2d 1596, 1601 (Fed.Cir.1998) (finding a system implementing a financial management structure satisfied §101 because it constituted a practical application of a mathematical algorithm by producing a useful, concrete and tangible result).

As provided above, the legal standard set forth by the Federal Circuit in *A&T Corp. v. Excel Communications, Inc.* for determining whether a claim is directed towards statutory subject matter is whether a claim can be applied in a practical application to ***produce a useful, concrete, and tangible result***. It is the result of the claims as applied in a practical application that is germane to the determination of whether the claims are directed towards statutory subject matter, not whether the underlying claims contain physical limitations (e.g., the computer readable medium embodiment cited by Examiner). The subject claims clearly satisfy this legal standard.

The subject claims determine whether a fault condition exists in executable code and provide information if a fault condition is determined to exist. Providing information based on determination of a fault condition is a useful, concrete and tangible result. By the logical principle of contraposition, if no information is provided, then the conclusion that no fault condition exists provides the further useful, concrete and tangible result of a measure of assurance that the executable code conforms to the code check framework of the invention.

In particular, independent claim 1 (as well independent claims 15, 17, 20, 24 and 25 that recite similar features) recites ***providing information if a fault condition is determined***. As a result of the claimed invention, executable object code faults are exposed by the static checker, which identified faults can then be corrected.

Accordingly, the claimed invention facilitates executable object code checking according to the provided framework. As described above, these claims recite an invention that produces a useful, concrete, and tangible result which clearly satisfies the standard set forth by the Federal Circuit for statutory subject matter under 35 U.S.C. § 101.

Regarding independent claims 22 and 23, these claims similarly recite a ***specification providing information to be employed to statically check the executable code***. For at least the reasons above, providing such information to facilitate static checking of executable code recites an invention that produces a useful, concrete, and tangible result which clearly satisfies the standard set forth by the Federal Circuit for statutory subject matter under 35 U.S.C. § 101.

In view of at least the foregoing, it is readily apparent that applicants' invention as recited in independent claims 1, 15, 17, 20, and 22-25 (and associated dependent claims 2-14, 16, 18-19, 21, and 23) are statutory subject matter and produce a useful, concrete, and tangible result. Accordingly, withdrawal of this rejection is respectfully requested in view of the foregoing comments.

III. Rejection of Claims 1-8 and 11-25 Under 35 U.S.C. §103(a)

Claims 1-8 and 11-25 stand rejected under 35 U.S.C. §103(a) as being unpatentable over disclosed art of record DeLine *et al*., "Enforcing High-Level Protocols in Low-Level Software," in view of Rickel *et al*., U.S. 5,854,924. Claims 1, 15, 17, 20, and 22-25 are the independent claims. Without conceding the propriety of the combination, applicants' representative respectfully request withdrawal of this rejection, because DeLine *et al*., alone or in combination with Rickel *et al*., does not teach or suggest each and every limitation of applicants' claimed invention.

> To reject claims in an application under §103, an examiner must establish a *prima facie* case of obviousness. A *prima facie* case of obviousness is established by a showing of three basic criteria. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation

> of success. Finally, the ***prior art reference (or references when combined) must teach or suggest all the claim limitations***. See MPEP §706.02(j). The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *See In re Vaeck,* 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

The present invention relates to static checking of object code, and, more particularly, to a system and method employing pre- and/or post- condition(s) specified at a source code level and persisted (e.g., in associated object code and/or a specification repository) facilitating static checking of the object code. The system and method are based, at least in part, upon a framework that employs rules for using an interface to be recorded as declarative specifications in an existing language. The executable code (*e.g.,* object code) with embedded specification(s) is then provided to a checker. The checker employs the specification to facilitate static checking of the object file. The checker provides information if a fault condition (*e.g.,* error(s)) is determined to exist. As a result, programming error(s) that could cause run-time exception(s) and that could escape traditional testing methods can be mitigated.

In contrast, the Vault programming language disclosed in Deline *et al*. allows a programmer to describe resource management protocols that the compiler can statically enforce through the use of an abstract global state of the program as it is compiled and type guards used by a type checker at compile-time. *See* page 1, Abstract and § 1, "Vault programming language provides . . . " *et seq*. "The global state, called the held-key set, consists of a set of keys, which are simply compile-time tokens representing run-time resources." *See* page 2, § 2.1, ¶ 1. Type guards can be used a programmer to specify domain-specific resource management protocols. *See* page 1, § 1, "Vault programming language provides . . . " *et seq*.

Although the Examiner concedes that Deline *et al*. does not expressly disclose that the received file having an embedded specification is an object file (Official Action p. 10), applicants' representative submits that Deline *et al*. teaches away from a persisted embedded specification in the object file as claimed. Notably, "[k]eys are ***purely compile-time entities that have no impact on runtime representations*** or execution

time." *See* page 2, § 2.1, ¶ 1 (emphasis added). Furthermore, "***[t]ype guards have no impact on run-time representation*** or execution time." *See* page 2, § 2.1, ¶ 2 (emphasis added).

Rickel, *et al.* cannot be said to cure this deficiency. Rickel, *et al.* merely discloses a static debugging tool for statically debugging a *representation of a binary program.* *See* Abstract. Such object file representations include an intermediate file created by a decompiler or otherwise, or a symbolic representation of the object file. See Col. 3, ll. 53-68 and Col. 2, ll. 29-36. The *representation* is then analyzed for errors by the Rickel, *et al.* system. Furthermore, Rickel *et al.* is silent with respect to any embedded specification information according applicants' invention. Although column 3, lines 18 – 26 and Column 4, lines 35 – 39 are cited for support that Rickel *et al.* discloses receiving an object file in executable code check system, applicants' representative submits that the *representation of the binary file* is not an object file according to applicants' invention.

Accordingly, the combination of Deline *et al.* with Rickel *et al.* fails to teach all the limitations of applications invention as claimed. Moreover, the resulting combination is not the applicant's invention – the resulting combination would consist of a debugger of representations of object code (Rickel *et al.*) that does not have persisted embedded specifications in the object code (Deline *et al.*)

For the avoidance of doubt, independent claim 1 (as well independent claims 17, 22, 24, and 25 that recite similar features) recites ***receives an object file having an embedded specification***. Because Deline *et al.* neither receives an object file nor has embedded specifications in the object file, and because Rickel *et al.* is silent with respect to embedded specifications, Rickel *et al.*, either alone or in combination with Deline *et al.* cannot be said to teach or suggest the ***receives an object file having an embedded specification.***

Regarding independent claims 15 and 23, these claims similarly recite ***static checking of the object file*** or ***statically check the executable code***. Because Deline *et al.* does not check an object file, and because Rickel *et al.* checks only an intermediate file representation of an object file or a symbolic representation of the object file, Rickel *et al.*, either alone or in combination with Deline *et al.* cannot be said to teach or suggest ***static checking of the object file*** or ***statically check the executable code***.

Regarding claim 20, Because Deline *et al*. neither receives nor checks executable code, and because in Rickel *et al*., an intermediate file representation of an object file or a symbolic representation of the object file is used for any analysis, Rickel *et al*., either alone or in combination with Deline *et al*. cannot be said to teach or suggest ***statically applying the specification to the executable code***.

Reconsideration and withdrawal of the rejection of claim 1, 15, 17, 20, and 22-25 (and associated dependent claims 2-14, 16, 18-19, 21, and 23) under 35 U.S.C. § 103(a) is respectfully requested in view of the comments above.

IV. Rejection of Claims 9 and 10 Under 35 U.S.C. § 103(a)

Claims 9 and 10 stand rejected under 35 U.S.C. § 103(a) as being unpatenable over DeLine *et al*. in view of Rickel, as applied to claim 1, and further in view of Alaluf, U.S. 2004/0230958. Claims 9 and 10 directly depend from claim 1. Without conceding the propriety of the combination, for at least the reasons set forth for claim 1 above, reconsideration and withdrawal of the rejection is respectfully requested.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP464US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/Himanshu S. Amin/
Himanshu S. Amin
Reg. No. 40,894

Amin, Turocy & Calvin, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731